

## **Day-4: Building Dynamic Frontend Components for My Marketplace**

### **1. Reusable Car Card Component**

- Created a reusable car card component to display car listings uniformly across the application.
- The component dynamically adjusts to show various car details, including name, type, price, image, and additional specifications.

### **2. Dynamic Car Detail Page**

- Developed a fully dynamic car detail page that displays specific details of the selected car.
- Ensured that clicking any car redirects users to a detail page showing information such as car name, type, price, transmission, fuel type, seats, and image.

### **3. Dynamic "Rent Now" Button**

- Integrated a dynamic "Rent Now" button on the car detail page.
- Clicking the button displays the car's price and a checkout form for booking.

### **4. Checkout Form with Validation**

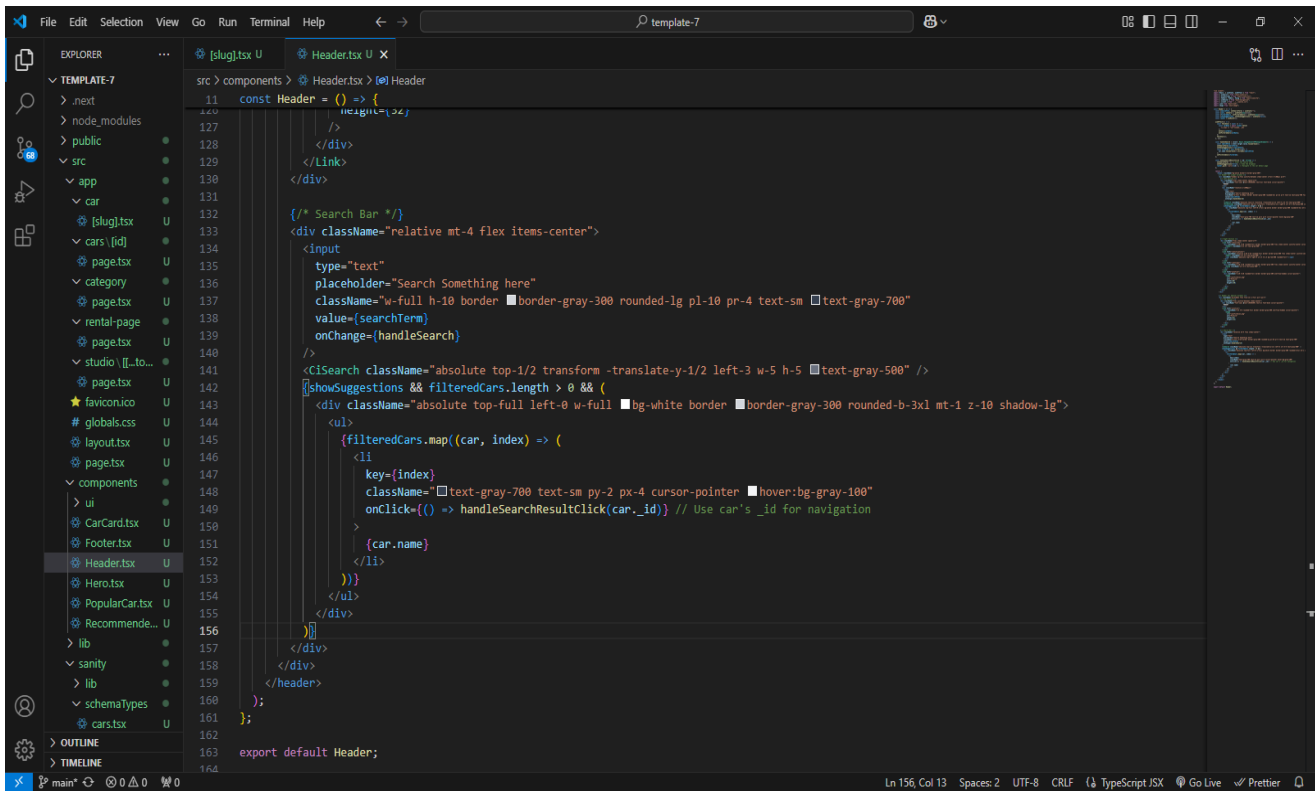
- Designed a user-friendly checkout form to collect customer details during the booking process.
- Added validation to ensure that all required fields are filled before proceeding, enhancing data accuracy and user experience.

### **5. Search Bar Functionality**

- Implemented a search bar to allow users to find cars by their name.
- Clicking the search result navigates users to the corresponding car detail page.
- Added an error message ("Car is not available") for cases where the entered car name does not match any record.

```
File Edit Selection View Go Run Terminal Help ← → template-7
[slug].tsx U Header.tsx U X
src > components > Header.tsx > Header
7 import { client } from "../../sanity.cli";
8 import Link from "next/link";
9 import Image from "next/image";
10
11 const Header = () => {
12   const [searchTerm, setSearchTerm] = useState("");
13   const [cars, setCars] = useState<any[]>([]);
14   const [filteredCars, setFilteredCars] = useState<any[]>([]);
15   const [showSuggestions, setShowSuggestions] = useState(false);
16   const router = useRouter();
17
18   useEffect(() => {
19     const fetchCars = async () => {
20       const carData = await client.fetch(
21         `*[_type == "car"]{name, _id}`
22       );
23       setCars(carData);
24       setFilteredCars(carData);
25     };
26     fetchCars();
27   }, []);
28
29   const handleSearch = (event: React.ChangeEvent<HTMLInputElement>) => {
30     const searchTerm = event.target.value.toLowerCase();
31     setSearchTerm(searchTerm);
32     setShowSuggestions(!searchTerm);
33     const filtered = cars.filter((car) =>
34       car.name.toLowerCase().includes(searchTerm)
35     );
36     setFilteredCars(filtered);
37   };
38
39   const handleSearchResultClick = (id: string) => {
40     setSearchTerm(""); // Clear the search input
41     setShowSuggestions(false); // Hide the dropdown
42     router.push(`/cars/${id}`); // Navigate to the car detail page
43   };
44
45   return (
```

```
File Edit Selection View Go Run Terminal Help ← → template-7
[slug].tsx U Header.tsx U X
src > components > Header.tsx > Header
11 const Header = () => {
45   return (
46     <header className="bg-white border-b border-gray-300">
47       { /* Header for Larger Screens */ }
48       <div className="hidden lg:flex justify-between items-center w-full h-[80px] px-6">
49         { /* Left Section */ }
50         <div className="flex items-center space-x-8">
51           <div className="font-sans text-[#3563E9] text-2xl font-bold cursor-pointer">
52             MORENT
53           </div>
54           <div className="relative w-[400px]">
55             <input
56               type="text"
57               placeholder="Search Something here"
58               className="w-full h-[36px] border border-gray-300 rounded-3xl pl-12 pr-4 text-sm text-gray-700 focus:outline-none"
59               value={searchTerm}
60               onChange={handleSearch}
61             />
62             <divSearch className="absolute top-1/2 transform -translate-y-1/2 left-4 w-5 h-5 text-gray-500" />
63             <divSettings className="absolute top-1/2 transform -translate-y-1/2 right-4 w-5 h-5 text-gray-500 cursor-pointer" />
64             {showSuggestions && filteredCars.length > 0 && (
65               <div className="absolute top-full left-0 w-full bg-white border border-gray-300 rounded-b-3xl mt-1 z-10 shadow-lg">
66                 <ul>
67                   {filteredCars.map((car, index) => {
68                     <li
69                       key={index}
70                       className="text-gray-700 text-sm py-2 px-4 cursor-pointer hover:bg-gray-100"
71                       onClick={() => handleSearchResultClick(car._id)}
72                     >
73                       {car.name}
74                     </li>
75                   )}
76                 </ul>
77               </div>
78             )}
79             </div>
80           </div>
81           { /* Right Section */ }
82         </div>
83       </header>
84     )
85   );
86 }
```



## 6. Fully Dynamic "Popular Cars" and "Recommended Cars" Components

- Developed a "Popular Cars" component to display a list of cars tagged as popular.
- Created a "Recommended Cars" component for displaying a curated selection of recommended vehicles.
- Both components are fully dynamic, ensuring that clicking any car in these sections takes the user to its respective detail page.

## 7. Dynamic Category Page

- Built a category page that shows all the cars in my sanity with the functionality of show more and show less button.
- Clicking on any car in a category redirects users to the respective car detail page dynamically.

## Conclusion

We have successfully developed our marketplace with the following key components:

- A reusable car card component used across multiple sections.
- Fully dynamic "Popular Cars" and "Recommended Cars" components for specialized car listings.
- A dynamic car detail component displaying individual car information.
- A fully dynamic rental page with a validated checkout form.
- A search bar component with error handling for unavailable car searches.
- A category page which show all the available cars.

This comprehensive development ensures an intuitive and seamless user experience across all pages of the marketplace.